

Quantum Oracles in Constant Depth with Measurement-Based Quantum Computation

Benoît Valiron

PPS, UMR 7126, Université Paris Diderot, Sorbonne Paris Cité, F-75205 Paris, France

(Dated: June 18, 2014)

This paper shows that, in measurement-based quantum computation, it is possible to write any quantum oracle implementing a classical function in constant depth. The result is shown through the equivalence between MBQC and the circuit model where arbitrary rotations along Z axis and unbounded fan-outs are elementary operations. A corollary of this result is that disjunction can be implemented exactly in constant-depth, answering an open question of Høyer and Špalek.

Proposed by Raussendorf and Briegel [6], the measurement-based quantum computational model is radically different from the circuit model. In the latter, the computation is performed on a set of quantum bit registers by successive applications of quantum gates. On the contrary, in the former the computation proceeds by adaptive one-qubit measurements performed on a *cluster state*, that is, a particular entangled multi-qubit state. The computation is encoded in the graph of entanglement, in the choice of basis for the measurements, and in their dependency graph.

In the measurement-based quantum computational model, the *depth* of the computation is the longest path in the dependency graph. Browne, Kashefi and Perdrix [2] show that this model is computationally equivalent to the circuit model where arbitrary rotations $R(\theta)$ around the Z axis

$$R(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix},$$

unbounded fan-outs and parity gates are taken as elementary gates. In particular, the depth-complexity of an algorithm is the same in both models, provided that *classical* unbounded parity gates are free.

Because of decoherence, the depth of an algorithm is a crucial limitation for quantum computation: in general, we want quantum algorithms to be as parallel as possible. Measurement-based quantum computation is a natural parallel computational paradigm and various works investigate its capabilities in terms of depth of computations [1, 2, 4]. In particular, if one considers *approximations* and not exact descriptions, several algorithms can be implemented in constant depth [4].

This paper presents a novel result with respect to exact descriptions: the fact that quantum oracles of the form $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$ can be *exactly* encoded in constant depth in measurement-based quantum computation.

As shown in Section I, it is clear that quantum circuits can easily do it with a suitable choice of elementary gates, provided that we allow circuits to have a width exponential on the size of the input. However, the fact measurement-based quantum computation can also do it has not been shown so far.

In order to prove this result, we use the equivalent representation in term of quantum circuits presented by

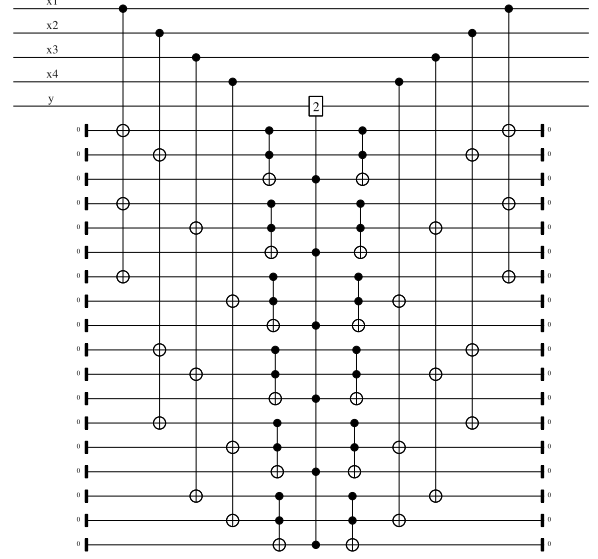


FIG. 1. Depth-5 oracle with multi-controlled CNOTs, fanouts and parity gates.

Browne and al. [2], and we generalize the decomposition of the Toffoli gate given by Selinger [7]. As a side effect, we also answer an open question of Høyer and Špalek [4]: there *is* a constant-depth exact circuit for the disjunction boolean operator.

I. NAIVE PARALLEL IMPLEMENTATION OF QUANTUM ORACLES

Consider a boolean function f on n inputs. This boolean function can always be written as

$$(x_1, \dots, x_n) \mapsto \bigoplus_{i=1}^N \bigwedge_{k \in K_i} x_k \quad (1)$$

where N is some natural number and the K_i 's some subsets of indices.

Provided that multi-controlled NOT-gates, unbounded fanouts and unbounded parity gates are available as elementary gates, this function can trivially be implemented

as a quantum oracle of the form

$$|x_1, \dots, x_n\rangle|y\rangle \mapsto |x_1, \dots, x_n\rangle|y \oplus f(x_1, \dots, x_n)\rangle,$$

in constant depth, as follows:

1. Allocate one block of ancillas for each K_i . The i -th block is of the size of K_i , plus one.
2. For each i , copy y and $\{x_k | k \in K_i\}$ to the corresponding block. This can be done in one step, with fanouts in parallel.
3. perform the conjunctions on each block, using multi-controlled NOT-gates. Again, this is one step.
4. Do the final XOR on the y gate using a parity gate (one step).
5. Undo the ancillas: multi-controlled NOT-gates, then fanouts (two steps).
6. Desallocate the ancillas

In total, not counting allocation and desallocation, the depth of the circuit is 5. As an example, the function

$$f(x_1, x_2, x_3, x_4) = (x_1 \wedge x_2) \oplus (x_1 \wedge x_3) \oplus (x_1 \wedge x_4) \oplus (x_2 \wedge x_3) \oplus (x_2 \wedge x_4) \oplus (x_3 \wedge x_4) \quad (2)$$

can be written as an oracle in depth 5 as in Figure 1 where $\boxed{2}$ stands for the parity gate. Note how the fanouts are indeed parallel. Now, any function f over an arbitrary number of input variables could be implemented with an oracle of the same shape, of depth 5. It is also easy to see how to extend this technique to the case of a boolean function f with more than one output.

The remainder of this paper is concerned with the implementation of this decomposition in MBQC, or equivalently [2] in a model of quantum circuit where unbounded fan-outs, Hadamard and rotations around the Z -axis are elementary gates.

II. A USEFUL EQUALITY

The main problem is the use of multi-controlled NOTs. In order to proceed with their decompositions using $R(\theta)$ -gates, we generalize the formula of Selinger relating conjunction of 3 boolean variables with XOR [7, Eq (5)] to Equation (3), relating the conjunction of n boolean variables with XOR.

As it is customary, we assimilate the boolean *false* with 0 and the boolean *true* with 1. The conjunction is simply the product, and we can transparently write boolean equations as equations over integers. With these conventions, one can show how to compute the conjunction of n booleans using XORs. This amounts to the Fourier spectra of the conjunction.

Lemma 1. *For all $n > 0$ and for any family $\{x_i\}_{i=1,\dots,n}$ of booleans, and if \mathcal{P}_i^n is the set of all subsets of $\{1 \dots n\}$ of size equal to i ,*

$$2^{n-1} \bigwedge_{i=1}^n x_i = \sum_{i=1}^n (-1)^{i-1} \sum_{K \in \mathcal{P}_i^n} \bigoplus_{k \in K} x_k \quad (3)$$

Proof. The proof is done by induction on n .

For $n = 1$, the equality is trivial.

For $n = 2$, the equality is

$$2x_1x_2 = x_1 + x_2 - x_1 \oplus x_2 \quad (4)$$

which can be shown correct by inspection of the 4 possible values for the pair (x_1, x_2) .

Now suppose that the equation is correct for $n \geq 2$, and consider the case $n + 1$:

$$2^n \bigwedge_{i=1}^{n+1} x_i = 2x_{n+1} \cdot 2^{n-1} \bigwedge_{i=1}^n x_i$$

which is, by induction hypothesis, equal to

$$2x_{n+1} \cdot \left(\sum_{i=1}^n (-1)^{i-1} \sum_{K \in \mathcal{P}_i^n} \bigoplus_{k \in K} x_k \right).$$

Expanding, this is equal to

$$\sum_{i=1}^n (-1)^{i-1} \sum_{K \in \mathcal{P}_i^n} 2x_{n+1} \bigoplus_{k \in K} x_k.$$

Using Eq. (4), we get

$$\sum_{i=1}^n (-1)^{i-1} \sum_{K \in \mathcal{P}_i^n} \left(x_{n+1} + \bigoplus_{k \in K} x_k - x_{n+1} \oplus \bigoplus_{k \in K} x_k \right).$$

One can then conclude using Lemma 4 (found in the appendix). \square

III. MULTI-CONTROLLED NOT GATES

Extending the technique presented in [7], together with auxiliary ancillas one can decompose any multi-controlled Z -gate as a circuit consisting of Clifford and $R(\theta)$ gates.

Lemma 2. *Any Z -gate controlled by $n \geq 2$ qubits can be written as a circuit consisting of (1) a sequence of CNOTs, (2) a list of $2^{n+1} - 1$ gates $R(\theta)$ in parallel, (3) a sequence of CNOTs.*

Proof. The proof is an adaptation of the one developed by Selinger [7], generalized to the n -ary case. Let T^n be the gate sending

$$|x_1 \dots x_n\rangle \mapsto (-1)^{x_1 \wedge \dots \wedge x_n} |x_1 \dots x_n\rangle$$

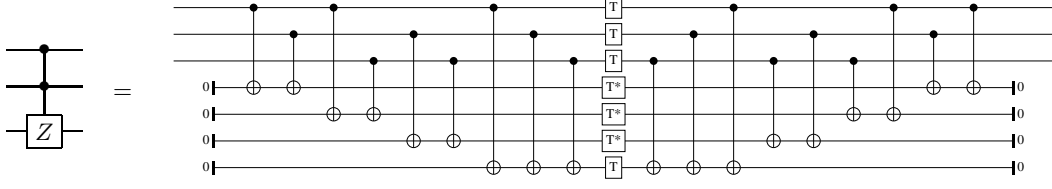


FIG. 2. Decomposition of the 2-controlled Z-gate.

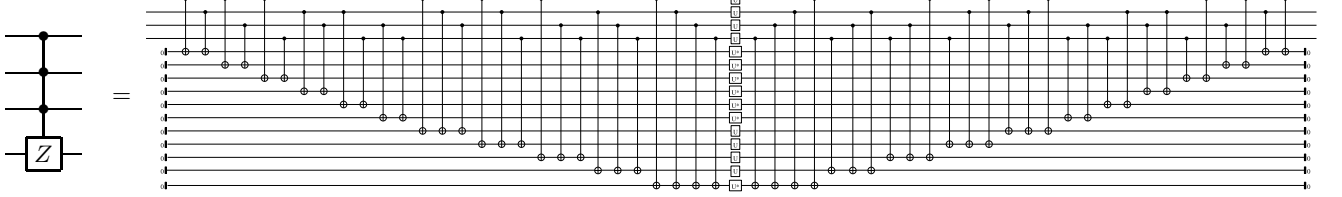


FIG. 3. Decomposition of the 3-controlled Z-gate.

with $n \geq 3$. This gate is a Z-gate controlled by $n - 1$ quantum bits. Thanks to Lemma 1, $(-1)^{x_1 \wedge \dots \wedge x_n}$ can be written as

$$\prod_{i=1}^n \prod_{K \in \mathcal{P}_i^n} \omega_n^{(-1)^{i-1} \oplus_{k \in K} x_k}$$

where $\omega_n = e^{\frac{i\pi}{2^n}}$. Therefore, the gate T^n can be implemented by applying $R(\frac{i\pi}{2^n-1})$ -gates and $R(\frac{-i\pi}{2^n-1})$ -gates to qubits in state $|\oplus_{k \in K} x_k\rangle$ where K are non-empty subsets of $\{1 \dots n\}$. One can construct and store these values using CNOT gates and $2^n - 1 - n$ ancillas: this allows the rotations gates to be set in parallel. The ancillas can then be reset to their original values, since the rotations around the Z-axis only change the phase. \square

As examples, we first show the 2-controlled Z-gate [7] in Figure 2: the T gate is $R(\frac{\pi}{4})$. We also show the case of the Z-gate controlled by 3 qubits in Figure 3, where the gate U is $R(\frac{\pi}{8})$. In both cases, the blocks of CNOTS are indeed made of pairwise commuting gates.

It is easy to see that in a given decomposition, each of the two blocks of CNOTs can be made of pairwise commuting gates: each block can then be encoded in constant-depth using unbounded fan-outs and parity gates [3, 5].

Therefore, because multi-controlled NOTs are two Hadamard away from multi-controlled Z-gates as shown in Figure 4, any multi-controlled NOT gate can be written in constant depth using arbitrary Z-rotations, Hadamard gates, unbounded fanouts and parity gates.

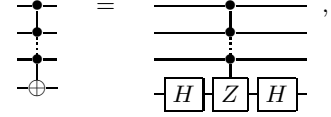


FIG. 4. Controlled NOTs and controlled Z-gates.

IV. QUANTUM ORACLES IN MBQC

Together with unbounded fanouts, Hadamard gates and arbitrary rotations along the Z axis and using the technique presented in Section I, one can therefore implement any boolean function in constant depth. Since constant-depth circuits using such gates can be implemented by constant-depth MBQC patterns [2], one concludes that quantum oracles can be implemented in constant depth in measurement-based quantum computation.

V. COMPLEXITY OF THE OVERALL SIZE

If the depth of the computation is constant, it is worth noting that in general the overall size is exponential with respect to the size of the input. Indeed, the width of the corresponding circuit corresponds to the sum of the numbers of subsets of the K_i in Eq. (1). For example, consider the function f as the conjunction, sending the vector (x_1, \dots, x_n) to $x_1 \wedge \dots \wedge x_n$. This is computed by a NOT-gate controlled by n quantum bits, which, from Lemma 2, can be represented by a quantum circuit consisting of $2^{n+1} - 1$ Z-rotations. The resulting MBQC

pattern is therefore exponential in n .

One can however recover a polynomial sized-pattern for Eq. (1) in the case where N is polynomial in n and when the size of the K_i is at most logarithmic in n . For example, the generalization of Eq. (2)

$$f(x_1, \dots, x_n) = \bigoplus_{i \neq j} x_i \wedge x_j$$

has a pattern representation of size polynomial on n .

VI. DISJUNCTION IN CONSTANT DEPTH.

We conclude this paper with a side comment, answering an open question. Høyer and Špalek have asked [4] whether the disjunction:

$$|x_1, \dots, x_n\rangle|y\rangle \longmapsto |x_1, \dots, x_n\rangle|y \oplus (x_1 \vee \dots \vee x_n)\rangle$$

can be implemented exactly by a constant-depth circuit. Using the results of the present paper, we can answer positively: using the fact that the disjunction of n variables $x_1 \vee \dots \vee x_n$ can be realized with a simple conjunction $\text{not}(\text{not } x_1 \wedge \dots \wedge \text{not } x_n)$, the requested circuit is essentially the decomposition of the multi-controlled NOT gate. However, note that the *size* of the circuit is exponential on n .

VII. ACKNOWLEDGMENTS

We would like to thank Simon Perdrix for enlightening discussions. This work was supported by the ANR project ANR-2010-BLAN-021301 LOGOI.

Appendix A: Auxiliary lemmas

In this appendix we recall two elementary results about binomial coefficients.

Let us write \mathcal{P}_i^n for the set of all subsets of $\{1 \dots n\}$ of size equal to i . If X is a set, let us write $\sharp X$ for the size of X . Note that $\sharp \mathcal{P}_i^n$ is the binomial coefficient $\binom{n}{i}$.

Lemma 3. *For all $n > 0$, for all $0 < i \leq n + 1$, the following equality holds: $\binom{n+1}{i} = \binom{n}{i-1} + \binom{n}{i}$.*

Proof. This is an easy corollary of the fact that the set \mathcal{P}_i^{n+1} is in fact $\{S \cup \{n+1\} \mid S \in \mathcal{P}_{i-1}^n\} \cup \mathcal{P}_i^n$. \square

Lemma 4. *For all $n > 0$, $\sum_{i=1}^n (-1)^{i-1} \binom{n}{i} = 1$.*

Proof. If $n = 1$, the lemma is true since there is only one element in a singleton. If $n > 1$, then using the previous lemma we deduce that

$$\sum_{i=1}^n (-1)^{i-1} \binom{n}{i} = \sum_{i=1}^n (-1)^{i-1} \left(\binom{n-1}{i-1} + \binom{n-1}{i} \right).$$

This is equal to

$$\binom{n-1}{0} + (-1)^{n-1} \binom{n-1}{n}.$$

The first element in the sum is 1, the second is 0. \square

-
- [1] A. Broadbent and E. Kashefi. Parallelizing quantum circuits. *Journal of Theoretical Computer Science*, 410(26), 2009.
 - [2] Dan Browne, Elham Kashefi, and Simon Perdrix. Computational depth complexity of measurement-based quantum computation. In *Proceeding of the Fifth Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2010)*, 2010.
 - [3] F. Green, S. Homer, C. Moore, and C. Pollett. Counting, fanout and the complexity of quantum ACC. *Quantum Information and Computation*, 2(1):35–65, 2002.
 - [4] Peter Høyer and Robert Špalek. Quantum fan-out is powerful. *Theory of Computing*, 1:81–103, 2005.
 - [5] C. Moore and M. Nilsson. Parallel quantum computation and quantum codes. *SIAM Journal on Computing*, 31(3): 799–815, 2002.
 - [6] R. Raussendorf and H. J. Briegel. Quantum computing via measurements only. *Physical Review Letters*, 86:5188–5191, 2001.
 - [7] Peter Selinger. Quantum circuits of T-depth one. *Physical Review A*, 87:042302, 2013.